A/B Testing Mekargo.id Website of Mekar PT Sampoerna Wirausaha with Bayesian Inference and Pymc3

Written by:

Name : Alvian Dwi Kurnianto, S.Kom.

MASTER DEGREE PROGRAM

GUNADARMA UNIVERSITY

Jakarta

2018

A/B Testing Mekargo.id Website of Mekar PT Sampoerna Wirausaha with Bayesian Inference and Pymc3

Written by:

Name : Alvian Dwi Kurnianto, S.Kom.

Student Id : 92216007

Supervisor : Dr. Setia Wirawan

Submitted as a Partial Fulllment of the Requirements for

Magister Degree of Information System Management

Concentrating on Information System Software

MASTER DEGREE PROGRAM

GUNADARMA UNIVERSITY

Jakarta

2018

ORIGINALITY STATEMENTS AND PUBLICATIONS

I, the undersigned below:

Name	:	**Alvian Dwi Kurnianto, S.Kom.**
NPM	:	**92216007**
Title	:	**"A/B Testing Mekargo.id Website of Mekar PT Sampoerna Wirausaha with Bayesian Inference and Pymc3"**
Date of Thesis Defence	:	April, 22 2018
Date of Graduation	:	

stated that this research is my own work and can be published entirely by Gunadarma University. All quotations of any kind have followed the rules and ethics of the day. Content and writing are the responsibility of the author. This statement is made in truth and with full awareness.

Jakarta, 16, April 2018

Alvian Dwi Kurnianto, S.Kom.

PAGE OF APPROVAL

Title	:	**"A/B Testing Mekargo.id Website of Mekar PT Sampoerna Wirausaha with Bayesian Inference and Pymc3"**
Name	:	**Alvian Dwi Kurnianto, S.Kom.**
NIM	:	**92216007**
Date of Thesis Defence	:	April, 22 2018
Date of Graduation	:	

Approved by:

Board of Advisors

.....

Dr. Setia Wirawan

(Chair Person)

.....

Prof. Dr. Yuhara Sukra., M.Sc.

(Member)

.....

Dr. Tb. Maulana Kusuma, S.Kom., MEngSc.

(Director)

ABSTRACT

Alvian Dwi Kurnianto, S.Kom.. 92216007
"A/B TESTING MEKARGO.ID WEBSITE OF MEKAR PT SAMPOERNA WIRAUSAHA WITH BAYESIAN INFERENCE AND PYMC3".

Mekar Go with address https://mekargo.id/ is a website to collect data of SMEs and employees who want to apply for a loan to Mekar. The current version of Mekar Go shows the conversion of a borrower who completes the data filling completion less than 10% which needs to be improved. To improve it, changing the order of data filling completion is developed with some different versions. All versions will have A / B testing to find out which version produced the most data conversion completion presentation. Data is collected by Google Analytics and analyzed using bayessian inference algorithm. Bayesian inference analysis is using pymc3, numpy, matplotlib, scipy, jupyter and Ipython tools with python programming language. Markov Chain Monte Carlo method is used to generate samples from a probability distribution. Metropolis-Hastings algorithm as the stochastic sampling technique. Bernoulli distribution as a discrete distribution. Stages of research methods conducted include: identifying problems, defining website measurement, developing a hypothesis, developing and testing page variants, and analyzing test results. The final result is the best version of all probabilities combinations between versions.

Keyword : A/B Testing, Bayesian Inference, Pymc3, Website, Google Analytics.

(xi+ 53)

References (2013-2018)

ACKNOWLEDGEMENTS

Praise and thanks raised to Allah SWT for all blessing and also utterance and good wishes sent to the prophet of Muhammad SAW so, with all the grants, author have finished this thesis, which is titled "A/B Testing Mekargo.id Website of Mekar PT Sampoerna Wirausaha with Bayesian Inference and Pymc3". This thesis is intended to complete the requirement to finish my study in the Business Information System Department, Gunadarma University. This writing would not have been possible without the support and encouragement from people around me. Therefore, I would like to say thank to:

1. Prof. Dr. Hj. E.S. Margianti, SE., MM, as Rector of Gunadarma University.
2. Prof. Dr. Yuhara Sukra MSc, as Coordinator of Postgraduate Program of Gunadarma University.
3. Prof. Dr. Dharma Tintri Ediraras, as Director of Postgraduate Program of Gunadarma University.
4. Dr. Yuhilza Hanum SSi, SKom, MEng, Diploma Program's Chief of Postgraduate Program of Gunadarma University.
5. Remi Senjaya, ST., MMSI, as Secretary of Sarjana Magister (Sarmag) Program.
6. Dr. Setia Wirawanas supervisor of this work, for her pa- tience has guided the author and examined the author's work in finish- ing this thesis.
7. My lovely wife Dyah Ayu Sukmawati for a lot of supports and kindness that always burns up my spirit to finish my thesis.

8. My beloved family, Father, Mother, Brother, Mother in Law, Father in Law, Brothers and Sisters in law for all supports, kindness, motivations and great affection that they have given.

9. PT Mekar Sampoerna Wirausaha, Dandi Rusli (CTO) and all others team members who gave permission to do research and help my development as a software developer.

10. All my friends, especially in Sarjana Magister (SarMag) SI - 03 for a lot of supports, sharings, jokes, and brother-sister hood.

Author realizes that there are no perfect things in the world, same as this work which may has many mistakes. Therefore, Author still looking forward to get some criticism and suggestion from anyone to make this work better in the future. Hopefully, this work can bring some advantages to readers.

.

Jakarta, April 2018

Alvian Dwi Kurnianto, S.Kom.

Contents

Originality Statements and Publications ii

Page of Approval iii

Abstract iv

Acknowledgements v

Contents vii

List of Figures x

List of Tables xi

1 INTRODUCTION 1

1.1 Background 1
1.2 Problem Identification 3
1.3 Scope of Problem 3
1.4 Research Purposes 3
1.5 Research Benefits 4

2 LITERATURE REVIEW 5

2.1 Python 5
2.2 Numpy 6
2.3 Matplotlib 6
2.4 Scipy 7
2.5 Ipython 8
2.6 Jupyter 8
2.7 Bayesian Inference 9

2.8 Markov Chain Monte Carlo 10
2.9 Pymc3 11
2.10 Theano 12
2.11 Google Analytics 13
2.12 A/B Testing 14
2.13 Previous Researches 15

3 RESEARCH METHODOLOGY 19
3.1 Research Object 19
3.2 Research Methodology 19
3.2.1 Identifying Problems 20
3.2.2 Defining Website Measurement 20
3.2.3 Developing a Hypothesis 22
3.2.3.1 How to Develop Hypothesis 22
3.2.4 Developing and Testing Page Variants 24
3.2.5 Analyzing Test Results 25
3.2.5.1 Bayessian Inference Algorithm 25
3.2.5.2 Bernoulli Distribution 25
3.2.5.3 Pymc3 26
3.2.5.4 Metropolis-Hastings Algorithm 26

4 RESULTS AND DISCUSSION 28
4.1 Identifying Problems 28
4.2 Defining Website Measurement 28
4.2.1 Business Objective 28
4.2.2 Website Goal 28
4.2.3 Key Performance Metric 29
4.2.4 Target Metric 29
4.3 Developing a Hypothesis 29
4.4 Developing and Testing Page Variants 29
4.4.1 Developing Page Variants 29
4.4.2 Testing Page Variants 32
4.5 Analyzing Test Results 34

5 CONCLUSION AND SUGGESTIONS 45
5.1 Conclusion 45
5.2 Suggestions 45

Bibliography **46**

Appendix **51**

Curriculum Vitae **53**

List of Figures

2.1 Python programming language usage in JOSS articles 6
2.2 Some examples of matplotlib chart 7
2.3 Jupyter notebook interface 9
2.4 Google Analytics dashboard 14
2.5 A/B Testing ilustration 15

3.1 Research Methodology 20
3.2 Website measurement 21

4.1 Graphic google analytics mekargo.id 33
4.2 Table google analytics mekargo.id 34
4.3 Import python library 34
4.4 Initializing sample 35
4.5 Calculating probability 35
4.6 Bernoulli distribution observation 36
4.7 Sum and mean of bernoulli distribution observation 37
4.8 Pymc probabilistic model 38
4.9 Pymc probabilistic model results 38
4.10 Posteriror distribution P_A 39
4.11 Posteriror distribution P_B 40
4.12 Posteriror distribution P_C 40
4.13 Posteriror distribution delta A-B 42
4.14 Posteriror distribution delta A-C 42
4.15 Posteriror distribution delta B-C 43
4.16 Final result combination of all variants 44

A.1 Letter of data and information permission request 51
A.2 Letter of data and information usage permission 52

List of Tables

2.1 Previous research 15

4.1 Data fields changes 30
4.2 Url definition 31
4.3 List of variables of A,B, and C version 41

Chapter 1

INTRODUCTION

1.1 Background

Small Medium Enterprise (SMEs) has an important role in encouraging the growth of the Indonesian economy. As presence of SMEs sector, unemployment due to the unabsorbed labor force in the workforce is reduced. SMEs sector has proven to be a pillar of a strong economy. The contribution of the SMEs sector in determining GDP and the country's foreign exchange earning sector is also unquestionable. Currently, SMEs have been the main agenda of Indonesia's economic development (Kemenkeu, 2015).

The most prominent issue of SMEs involves providing business financing or business capital. Business capital needs are felt when someone wants to start a new business. As a result, usually when the motivation is strong, someone will still start a business with makeshift capital. In the business that has been running, the capital remains a further constraint to develop. Problems facing SMEs concerning the ability of access to finance, market access and marketing, management of small business management and access to information. The difficulties of SMEs to access capital resources because of limited information and ability to penetrate the source of capital. Despite the choice of capital resources are numerous and varied. Bank institutions are the largest source of capital that can be utilized by small business actors. However, to partner with banks, small businesses are required to present business proposals that are feasible or feasible and profitable (Wuisan, 2017).

Mekar (PT Mekar Investama Sampoerna) is fully supported by the Put-

era Sampoerna Foundation. Mekar aims to improve access to finance to small businesses that have a positive economic and social impact in Indonesia. We do this by connecting funders with businesses needing finance. To enable this we partner closely with (non-bank) financial institutions which are present all over Indonesia (Mekar, 2017). Mekar provides solutions to SMEs and other Consumers to get Financial Services Access or Capital in Indonesia. Some products owned by Mekar include Mekar.id main website, chat bot and website of Mekar Go. Mekar Go Website with web address https://mekargo.id/ is a website to collect data of SMEs and employees who want to apply for a loan to Mekar. The data collected on the current version of Mekar Go Website shows the conversion of a borrower who completes the data filling from start to finish less than 10%. This conversion percentage is too small and needs to be improved.

To improve the percentage of data completion conversions, the marketing team proposes changing the order of data filling. The change of the data filling completion is made into two new versions referred to as version B and version C, and the current version is referred to as version A. All three versions will have A / B testing to find out which version produced the most data conversion completion presentation. A / B testing is a process used in marketing to isolate and test factors that affect the performance of which version of a product meets marketing criteria (Dixon et al., 2015). A / B testing is widely used for website optimization: two versions of a web page, say A and B, empirically compared to presented to the user. Each user only sees one of two versions, and the goal is to determine which version is better. The goal to be achieved in general is to determine which web page has the highest conversion rate (the possibility of the user actually being a customer) by receiving feedback from the user (Kaufmann et al., 2014). In this research, collecting user feedback data is done with Google Analytics tools. Google Analytics is an analytics tool created by Google, used to measure website performance and user behavior while visiting the website (Yang and Perrin, 2014). Once the data collected by Google Analytics is met the expectation, a data analysis process using the bayessian inference and pymc3 as the tools. Pymc3 is a probabilistic programming framework with bayessian inference written in the python programming language (Salvatier et al., 2016).

1.2 Problem Identification

The problem discussed in this research is to determine which version of mekargo.id website at PT Mekar Sampoerna Wirausaha with A / B testing which become the best version with bayesian inference method with pymc3 tools.

1.3 Scope of Problem

The problem limitation of this research is as follows:

1. The object of research is the website mekargo.id.
2. The research was conducted at PT Mekar Sampoerna Wirausaha.
3. Study time between August 2017 to January 2018.
4. Data collection method for A / B testing techniques on mekargo.id website with the help of Google Analytics tools.
5. The generated data from A / B testing is analyzed by Bayesian inference analysis using pymc3, numpy, matplotlib, scipy, jupyter and Ipython tools with python programming language.
6. Mekargo.id website features that studied is a feature of charging data borrower.
7. Mekargo.id website feature that is not discussed is a static page feature that contains the home page, about us, terms and conditions, privacy policy and contact us.

1.4 Research Purposes

The purpose of this research is to know the version of A / B testing which is the best version of mekargo.id website at PT Mekar Sampoerna Entrepreneurship with bayesian inference method with pymc3.

1.5 Research Benefits

The benefits of this research are:

1. For the researcher, be a guide or research reference in the field of information system in A / B testing with bayessian inference method.
2. For other researchers, as a literature source for A / B testing with bayessian inference method.
3. For the company under study, provide an alternative to solving the problem of choosing the best version with A / B testing.

Chapter 2

LITERATURE REVIEW

2.1 Python

Python is a dynamically typed programming language that has a focus on ease of use and readability. Due in part to this focus, it has become a popular language for scientific computing and data science, with a broad ecosystem of libraries (Meurer et al., 2017). Python syntax is designed very readable, which is important for software development. Every computer program is written only once, but read and revised many times, often by many people. Being readable also makes it easier to learn and remember, hence more writeable. Compared with other popular languages, Python has a gentle learning curve that makes developer more productive, yet it has depths that can be explored and gain expertise (Lubanovic, 2014).

People use python because have these several advantages: software quality, developer productivity, program portability, support libraries, component integration, and enjoyment (Lutz, 2013). For application development, Python has advadtages of ease for development and prototyping (Vohl et al., 2016). Python is a multi-platform, general-purpose programming language that can run on Windows, Linux/Unix, and Mac OS X, and has been ported to Java and .NET virtual machines as well. It has a powerful standard library. In addition, it has many libraries for data analysis such as Numpy, Pandas, Matplotlib, PyMongo, and Scikit (Vo et al., 2015). Python is the most popular programming languages in the scientific domains (Rougier et al., 2017). The frequency of programming languages appearing in Journal of Open Source Software (JOSS) articles show that Python appears the most with over half

of published articles in total 54 (Smith et al., 2018). The statistic of Python programming language that appeared in JOSS is shown by figure 2.1.

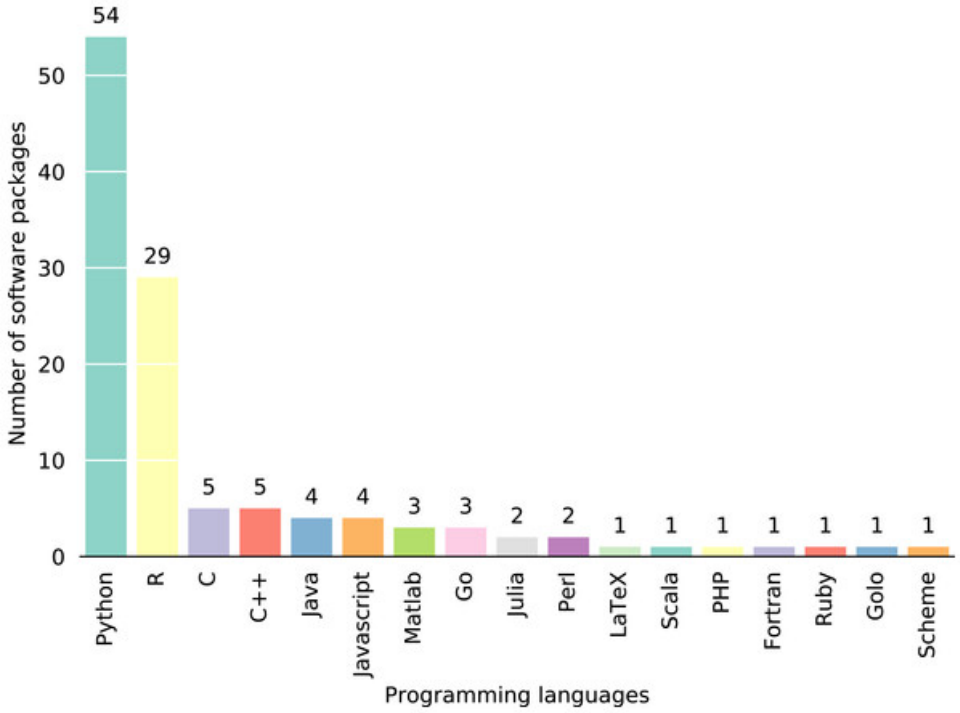


Figure 2.1: Python programming language usage in JOSS articles
Source: Smith et al., 2018

2.2 Numpy

Numpy is a fundamental package that is extremely useful for scientific computing, and contains among other things a powerful N-dimensional array object (Strickland et al., 2014). The Numpy package, which comprises the Numpy array as well as a set of accompanying mathematical functions, has found wide-spread adoption in academia, national laboratories, and industry, with applications ranging from gaming to space exploration (Van Der Walt et al., 2016).

2.3 Matplotlib

Matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments

across platforms. Matplotlib can be used in python scripts, the python and ipython shell, web application servers, and six graphical user interface toolkits (Hunter et al., 2014). Matplotlib provides both a very quick way to visualize data from Python and publication-quality figures in many formats: line plots, contour plots, scatter plots, and basemap plots. It comes with a set of default settings, but allows customization of all kinds of properties. However, we can easily create our chart with the defaults of almost every property in Matplotlib (Vo et al., 2015). Some of charts example that can be generated by matplotlib are ilustrated by figure 2.2.

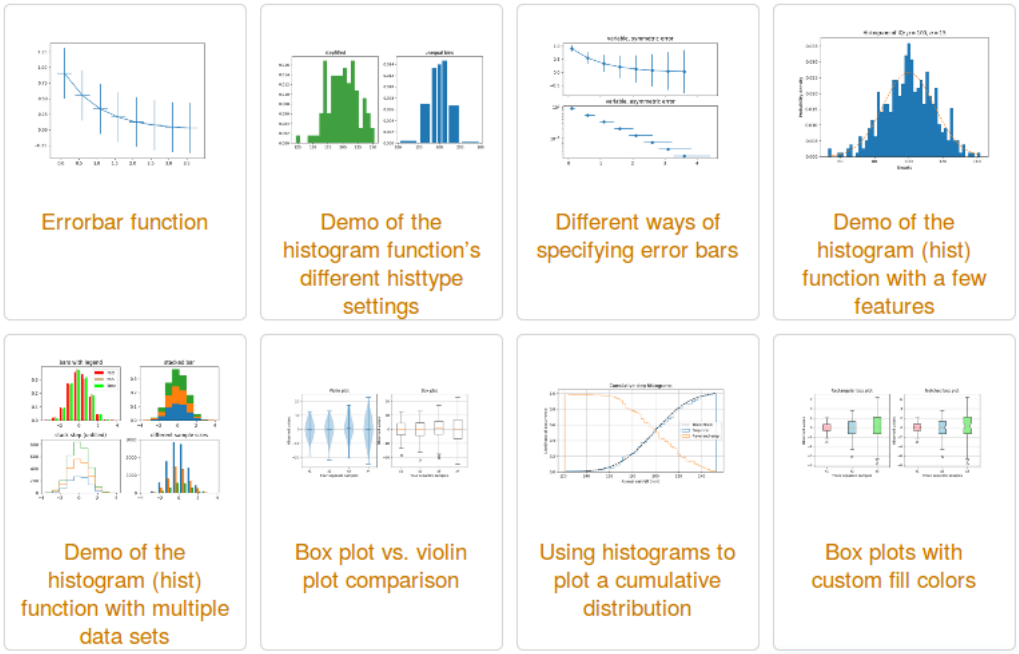


Figure 2.2: Some examples of matplotlib chart
Source: Hunter et al., 2014

2.4 Scipy

Scipy (pronounced “Sigh Pie”) is a Python-based ecosystem of open-source software for mathematics, science, and engineering. Scipy ecosystem includes general and specialised tools for data management and computation, productive experimentation and high-performance computing. The SciPy library, a collection of numerical algorithms and domain-specific toolboxes, including signal processing, optimization, statistics and much more (Jones et al., 2014).

2.5 Ipython

Ipython is an interactive browser-based environment where developer can combine code execution, text, mathematics, plots, and rich media into a single document. Originally designed for use as an electronic lab notebook for computational science, it is increasingly being used in teaching as well, and a rich ecosystem of open source plugins and extensions for teaching is growing around it (Wilson et al., 2014). IPython has provided terminal-based tools for interactive computing in Python since 2001. The notebook document format and multi-process architecture introduced in 2011 have expanded the applicable scope of IPython into teaching, presenting, and sharing computational work, in addition to interactive exploration (Ragan-Kelley et al., 2014).

2.6 Jupyter

Jupyter Notebook, provides a tool to create and share web pages with text, charts, and Python code in a special format. Often, the notebooks are used as an educational tool, or to demonstrate Python software. We can import or export notebooks either from plain Python code or from the special notebook format. The notebooks can be run locally, or we can make them available online by running a dedicated notebook server (Fandango and Idris, 2017).

Jupyter Notebook is accessed through a modern web browser. This makes it practical to use the same interface running locally like a desktop application, or running on a remote server. In the latter case, the only software the user needs locally is a web browser; so, for instance, a teacher can set up the software on a server and easily give students access. The notebook files it creates are a simple, documented JSON format, with the extension '.ipynb'. It is simple to write other software tools which access and manipulate these files (Kluyver et al., 2016). Jupyter notebook interface is shown by figure 2.3.

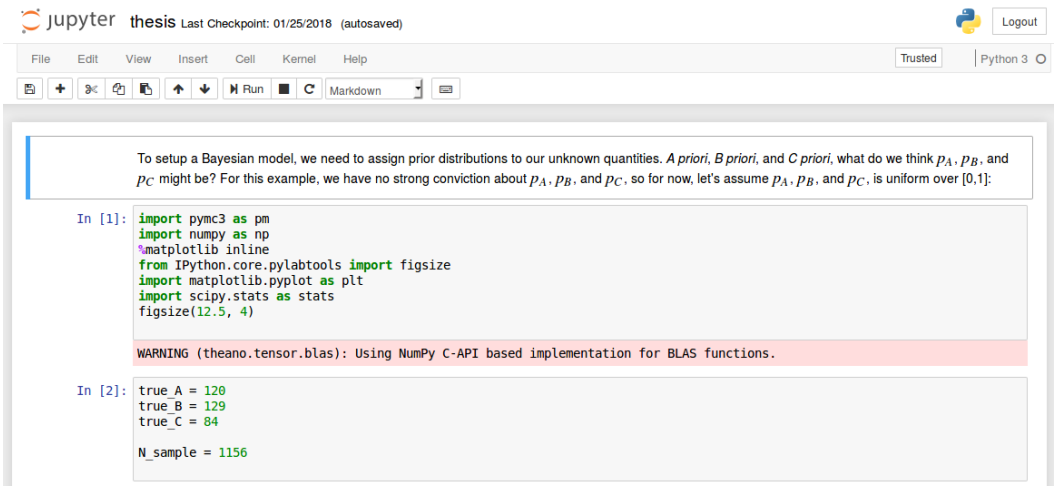


Figure 2.3: Jupyter notebook interface

2.7 Bayesian Inference

Bayesian inference is the process of fitting a probability model to a set of data and summarizing the result by a probability distribution on the parameters of the model and on unobserved quantities such as predictions for new observations (Gelman et al., 2014). Bayesian inference differs from more traditional statistical inference by preserving uncertainty. The Bayesian worldview interprets probability as measure of believability in an event, that is, how confident we are in an event occurring. In fact, we will see in a moment that this is the natural interpretation of probability (Davidson-Pilon, 2015).

Bayesian inference departs from the approach to statistical inference described in many textbooks, which is based on a retrospective evaluation of the procedure used to estimate θ (or $\tilde{y}$) over the distribution of possible y values conditional on the true unknown value of θ. Despite this difference, it will be seen that in many simple analyses, superficially similar conclusions result from the two approaches to statistical inference. However, analyses obtained using Bayesian methods can be easily extended to more complex problems (Gelman et al., 2014).

The Bayesian modeling paradigm involves setting up a probabilistic model describing how data was gen- erated, assigning prior distributions over unknown model parameters, and then calculating a posterior distribution over these parameters. In practice, this posterior distribution is often intractable to compute exactly except for the simplest models. This often requires one

to resort to approximate posterior inference algorithms based on Monte-Carlo sampling or Variational Inference (VI). Fortunately, a lot of progress has been made recently in this area including state-of-the-art algorithms such as Hamiltonian Monte Carlo sampling (HMC), the No-U-Turn Sampler (NUTS), Automatic Differentiation VI (ADVI) and Black Box VI. These algorithms are applica- ble to a wide class of models and are readily available in popular Probabilistic Programming languages such as Stan, Edward, and PyMC3 (Pourzanjani et al., 2017). Bayesian inference had been applied in many areas such as automatic machine-learning hyperparameter optimization, A/B testing or recommender systems, among others (Jiménez and Ginebra, 2017).

Some advantages of the Bayesian approach are: that it provides a transparent framework for inference; secondly, it is flexible and often provides an elegant and practical solution to inference arising from very complex statistical models. Note, in particular, that Bayesian methods make no formal distinction between estimation oand prediction, and in this way naturally incorporate parameter uncertainty into predictive inference. Because, obtaining the sample can itself be a major challenge (Taylor et al., 2015).

2.8 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) method is used to generate samples from a probability distribution. It is widely used nowadays in many aspects of optimization and numerical integration, and is specially suitable for being used in sampling from posterior distributions in Bayesian inference. A key issue in MCMC is whether the chain has converged and is actually sampling from the target distribution. There is not a single infallible test of convergence, but many formal and informal ways to assess non-convergence (Fernández-i Marín et al., 2016).

Markov chain Monte Carlo is a general method based on drawing values of θ from approximate distributions and then correcting those draws to better approximate the target posterior distribution, $p(\theta|y)$. The sampling is done sequentially, with the distribution of the sampled draws depending on the last value drawn; hence, the draws form a Markov chain. (As defined in probability theory, a Markov chain is a sequence of random variables θ^1 , θ^2 , . . ., for which, for any t, the distribution of θ^t given all previous θ's

depends only on the most recent value, θ^{t-1} .) The key to the method's success, however, is not the Markov property but rather that the approximate distributions are improved at each step in the simulation, in the sense of converging to the target distribution (Gelman et al., 2014).

2.9 Pymc3

PyMC3 is a new, open-source probabilistic programming framework with an intuitive and readable, yet powerful, syntax that is close to the natural syntax statisticians use to describe models. It features next-generation Markov chain Monte Carlo (MCMC) sampling algorithms (Salvatier et al., 2016). Running PyMC3 requires a working Python interpreter (Python Software Foundation, 2010), either version 2.7 (or more recent) or 3.4 (or more recent). PyMC3 can be installed using 'pip': pip install pymc3. PyMC3 depends on several third-party Python packages which will be automatically installed when installing via pip (Salvatier et al., 2016).

The majority of the heavy lifting done by PyMC3 is taken care of with the theano package. The notation in theano is remarkably similar to NumPy. It also supports many of the familiar computational elements of NumPy. However, while NumPy directly executes computations, e.g. when run a + b, theano instead builds up a "compute graph" that tracks that to perform the + operation on the elements a and b. Only when eval() a theano expression does the computation take place (i.e. theano is lazy evaluated). Once the compute graph is built, we can perform all kinds of mathematical optimizations (e.g. simplifications), compute gradients via autodiff, compile the entire graph to C to run at machine speed, and also compile it to run on the GPU. PyMC3 is basically a collection of theano symbolic expressions for various probability distributions that are combined to one big compute graph making up the whole model log probability, and a collection of inference algorithms that use that graph to compute probabilities and gradients (Davidson-Pilon, 2015).

2.10 Theano

Theano is a Python library that allows to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. Since its introduction in it has been one of the most used CPU and GPU mathematical compilers – especially in the machine learning community and has shown steady performance improvements. Theano is being actively and continuously developed since 2008, multiple frameworks have been built on top of it and it has been used to produce many state-of-the-art machine learning models. Theano allows a user to symbolically define mathematical expressions and have them compiled in a highly optimized fashion either on CPUs or GPUs (the latter using CUDA) 1 , just by modifying a configuration flag. Furthermore, Theano can automatically compute symbolic differentiation of complex expressions, ignore the variables that are not required to compute the final output, reuse partial results to avoid redundant computations, apply mathematical simplifications, compute operations in place when possible to minimize the memory usage, and apply numerical stability optimization to overcome or minimize the error due to hardware approximations. To achieve this, the mathematical expressions defined by the user are stored as a graph of variables and operations, that is pruned and optimized at compilation time.

The interface to Theano is Python, a powerful and flexible language that allows for rapid prototyping and provides a fast and easy way to interact with the data. The downside of Python is its interpreter, that is in many cases a poor engine for executing mathematical calculations both in terms of memory usage and speed. Theano overcomes this limitation, by exploiting the compactness and ductility of the Python language and combining them with a fast and optimized computation engine. Theano's API mimics NumPy, a widely adopted Python library that provides an n-dimensional array data type and many functions for indexing, reshaping, and performing elementary computations (exp, log, sin, etc.) on entire arrays at once. This allows Python users to rapidly switch to Theano using a familiar syntax and set of instructions – extended with advanced features, such as automatic gradient computation, numerical stability improvements and optimization – and generate a high-performance code for CPU as well as for GPU, without requiring changes to the user code. Theano has also been designed for easy and fast

extensibility through the definition of custom graph expressions written in Python, C++, or CUDA (Al-Rfou et al., 2016).

2.11 Google Analytics

Google Analytics (GA) is an easy-to-use tool to measure activity on a website. A basic setup might take as little as a few minutes, and many of the standard reports are quite accessible and understandable without any special training or prior knowledge of web analytics. Because of this, many users jump into GA without knowing much about its underpinnings—how the data is structured and gathered—and that's fine for the basics. But eventually, users can outgrow this intuitive understanding of GA and its data, and need deeper insight into how it works and what it can do (Weber, 2015).

Google Analytics is free, and it is always increasing in power, to rival and in some cases exceed the performance of the "paid" tools (Kelsey, 2017). Google Analytics has another advantages such as has tremendous features, and ease of use. Unlike other Web analytics tools that use server log files, Google collects information by inserting simple Javascript codes into Web pages. The advantage of this method is that Google Analytics can capture technical and demographic information that log files do not normally provide, such as the user's browser, operating system, screen size, resolution, and so on (Yang and Perrin, 2014). The example of Google Analytics dashboard is shown by figure 2.4.

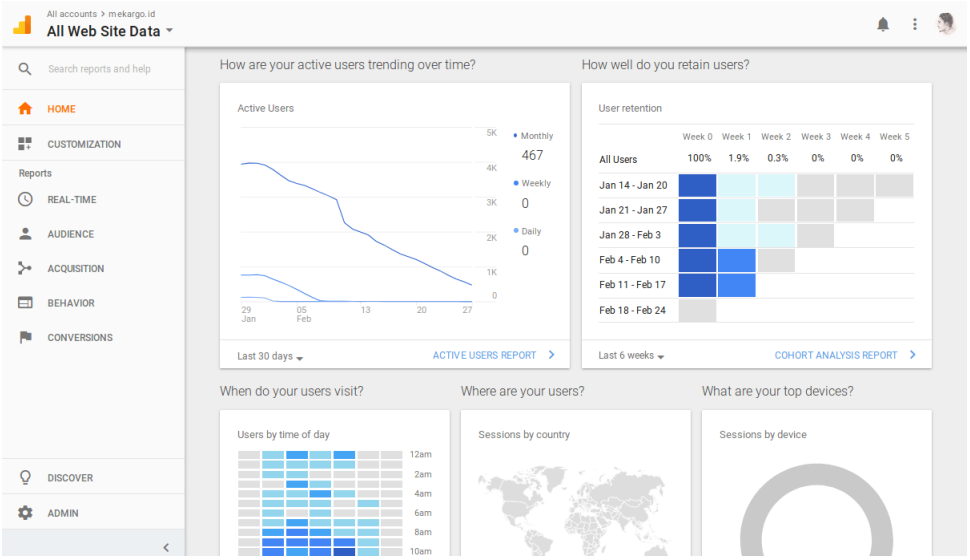


Figure 2.4: Google Analytics dashboard

2.12 A/B Testing

A/B testing has widely become the standard controlled experimentation framework for network driven companies like Facebook, LinkedIn, Twitter, Google, and Yahoo (Wilson and Uminsky, 2017). A/B testing is ubiquitous within the machine learning and data science operations of internet companies. Generically, the idea is to perform a statistical test of the hypothesis that a new feature is better than the existing platform. For example, it results in higher revenue (Goldberg and Johndrow, 2017).

A/B testing started to be used in the late 1990s with the growth of the Internet. Many large companies run thousands to tens of thousands of A/B testing each year testing user interface (UI) changes, enhancements to algorithms (search, ads, personalization, recommendation, etc.), changes to apps, content management system, etc. In an A/B testing, users are randomly split between the variants (e.g., the two different ads layouts) in a persistent manner (a user receives the same experience in multiple visits). Their interactions with the site are instrumented and key metrics computed (Kohavi and Longbotham, 2017).

In A/B testing, one has a proposed new version of a software platform and wants to decide whether or not to ship the new version. The classical way of conceiving of this problem is the following. We divide users into two groups: treatment and control. We then roll out the proposed update to the

treatment group while leaving the control group with the current version. Using data gathered from this randomized trial, we then ask whether the new version performed "better" with respect to some metric (Goldberg and Johndrow, 2017). A/B testing can be ilustrated by figure 2.5.

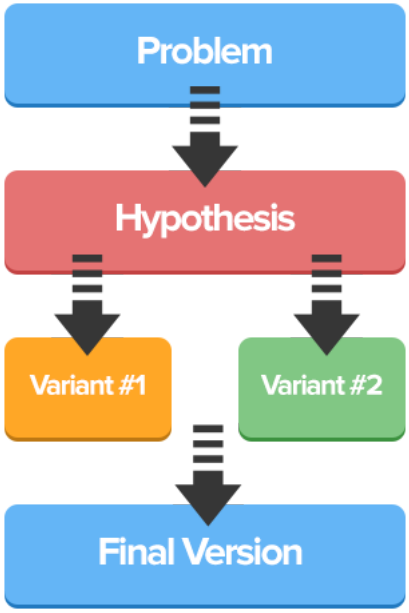


Figure 2.5: A/B Testing ilustration
Source: Omniconvert, 2017

2.13 Previous Researches

Table 2.1 contains previous researches on A/B testing. It contains number, title of the article, authors of the article, and strength and of the article from author point of view. There are six articles that author discuss. The point of view is composed from the mathematical concept explanation, application example, and suplementary material of the articles.

Table 2.1: Previous research

No	Article	Authors	Strength	Weakness

No	Article	Authors	Strength	Weakness
1	A Decision Theoretic Approach to A/B Testing(Goldberg and Johndrow, 2017)	David Goldberg (eBay) and James E. Johndrow (Stanford University)	Great on Mathematical concept explanation	Lack of application example
2	Always Valid Inference: Bringing Sequential Analysis to A/B Testing(Johari et al., 2015)	Ramesh Johari, Leo Pekelis and David J. Walsh	Great on real world example and Mathematical concept explanation. Also contains Supplementary Material	There is no proof of theorem explanation
3	Online Controlled Experiments and A/B Testing(Kohavi and Longbotham, 2017)	Ron Kohavi (Microsoft) and Roger Longbotham (Microsoft)	Great on real world example	Lack of Mathematical concept explanation

No	Article	Authors	Strength	Weakness
4	Online Controlled Experiments at Large Scale (Kohavi et al., 2013)	Ron Kohavi (Microsoft), Alex Deng (Microsoft), Brian Frasca (Microsoft), Toby Walker (Microsoft), Ya Xu (Microsoft), and Nils Pohlmann (Microsoft)	Great on real world example	Lack of Mathematical concept explanation
5	On the Complexity of A/B Testing (Kaufmann et al., 2014)	Emilie Kaufmann (LTCI, Télécom ParisTech & CNRS) , Olivier Cappé (LTCI, Télécom ParisTech & CNRS), and Aurélien Garivier (Institut de Mathématiques de Toulouse, Université Paul Sabatier)	Great on Mathematical concept explanation	Lack of application example

No	Article	Authors	Strength	Weakness
6	The Power of A/B Testing under Interference (Wilson and Uminsky, 2017)	James D. Wilson (Department of Mathematics and Statistics University of San Francisco) and David T. Uminsky (Department of Mathematics and Statistics University of San Francisco)	Great on real world example and Mathematical concept explanation. Also, completed with proof of theorem explanation	There is no Supplementary Material

Chapter 3

RESEARCH METHODOLOGY

3.1 Research Object

The time of the study was conducted between August 2017 and January 2018. The object of this research is Mekar's Mekargo.id website. Mekar through Mekar Go provide solutions in assisting SMEs and other consumers to gain access to financial services or capital access in Indonesia. Mekar provides easy access through portal Mekar Go to financial institutions such as Banks, finance companies, and Rural Banks in improving the quality of service to the community easily, safely and quality.

The feature studied in this research is the registration feature for the borrower. The data used is the url address that the registrant visits for each version of A / B testing. Data collection is done with google analytics tools. The data obtained will then be analyzed by bayesian inference and pymc3 as tool.

3.2 Research Methodology

Research methods include: identifying problems, defining website measurement, developing a hypothesis, developing and testing page variants, and analyzing test results. The description of the steps of the research method can be illustrated by figure3.1.

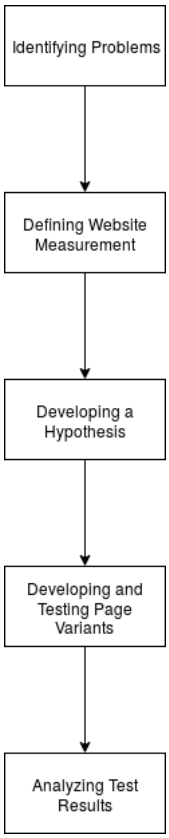


Figure 3.1: Research Methodology

3.2.1 Identifying Problems

Issues identified are issues related to the website to be performed A / B testing. The issues raised are usually related to business, marketing, or convertion rate. Depending on what goals to be achieved on making A / B testing on the website. The problem described is usually because there is something that arises that does not match the goal. In addition, also described the current conditions of the website.

3.2.2 Defining Website Measurement

Website measurement is a measurement done to provide a goal to achieve the desired A / B testing objectives. Website measurement consists of several business indicator points outlined from the general to the details. This indicator point consists of business objectives, website goals, key performance metrics (KPIs), and target metrics. Given this description, it will be more clear what the desired results of A / B testing are. Thus, all members of

the team involved can unify the vision in performing this A / B testing. The diagram can be illustrated in figure3.2.

Figure 3.2: Website measurement
Source: Omniconvert, 2017

The description of the measurement website indicator points are as follows :

- **Business objectives** are the main targets to be achieved in making A / B testing. For example increasing the number of sales, increasing the number of users who register, or increase visitors on page views. This point depends on the type of business being run and what stage is the problem.

- **Website goals** are what actions to achieve or improve on a single website feature that can meet the objectives of a business objective. If want to increase the number of sales, then the goal website that must be achieved is an increase feature add items to the chart until making payment. Website goals depend on the business objective points to be achieved.

- **Key performance metrics** are measures that guide the success of what to be achieved. For example want to increase sales, then the size used is how many items entered into the chart, how many items on the chart made checkout payment, and so forth.

- **Target metrics** are how many target numbers are targeted for success of each key performance metric accompanied by a timeframe that becomes the limit. For example, target the quantity of goods inserted into a chart of at least 1,000,000 items per day. That is one example for ecommerce companies that already bersar. The targeted number of numbers depends on the current state. The most important is the increase.

3.2.3 Developing a Hypothesis

Many of the variables specifically require to bring data to the table to prioritize hypotheses.

- Is it addressing an issue discovered via user testing?
- Is it addressing an issue discovered via qualitative feedback (surveys, polls, interviews)?
- Is the hypothesis supported by mouse tracking heat maps or eye tracking?
- Is it addressing insights found via digital analytics?

Having weekly discussions on tests with these 4 questions asked from everyone will quickly make people stop relying on just opinions. There are also bounds placed on Ease of implementation by bracketing answers according to the estimated time.

A good hypothesis:

1. Is testable – hypothesis is measurable, so that it can be used in testing has a goal of solving conversion problems – Split testing is done to solve specific conversion problems
2. Gains market insights – Besides increasing conversion rates, split testing will give information about clients.
3. A well-articulated hypothesis will let split testing results give information about customers.

After the problem have defined and hypothesis is articulated, A/B testing can be started to come up with specific split-testing variations. Clearly defining hypothesis will help yo come up with testing variations that give meaningful results (CXL, 2017).

3.2.3.1 How to Develop Hypothesis

Case Studies : Conversion rate optimization relies heavily on case studies to guide best practices and develop new hypothesis. While it's not always true, case studies often represent solutions that might work in a given situation. Since at the hypothesis stage, the job is to collect as many plausible

solutions as many as can, digging through case studies can be extremely useful. However, not every case study will actually yield actionable insights to specific case. When evaluating a case study, consider these issues:

- **Relevancy**: Is the case study describing a problem that is relevant to particular problem?
- **Audience overlap**: Is the case study about a website that has a similar audience?
- **Accuracy**: Are the results from the case study accurate enough? Was the testing carried out to a high degree of confidence? Far too many times, many companies promoting case studies that are not statistically accurate.
- **Recency**: Are the findings from the case study recent enough? Have new trends impacted these results? What worked for a brand in 2006 might not hold true in 2016.
- **Scope**: Plenty of case studies focus on "small wins" or micro-conversions, such as the CTR for a button. Such results might look good, but unless they improve target metrics, they are effectively meaningless. Always consider the scope of the study before borrowing its learnings in the developed tests.

Best Practices: Best practices are just that - a practice assumed to be correct by default in any non-specific case. For example, placing navigation bar at the top of the page is a "best practice". Best practices spring from three things:

- **Convention**: Some design choices are made just because that's the way things have always been done. In most cases, it's better to follow conventions since customers are already used to it. Breaking convention might cause confusion.
- **Theory**: Some design choices spring from theory. For example, psychological theories state that we are likely to equate strong social proof with trustworthiness. Hence, including reviews, testimonials, etc. has now become a "best practice".

- **Testing**: When usability and CRO tests yield the same insights consistently, they often become best practices. For instance, testing shows that descriptive CTAs (such as “Download eBook”) regularly perform better than plain CTAs (such as “Submit”). This can now be said to be a “best practice”.

Theory: Conversion rate optimization sits at the intersection of sales, design and psychology. As such, any hypothesis come up with will likely involve theories from each of these fields. To give an example, good UI/UX design relies heavily on Gestalt psychology (such as the Law of Proximity and the Law of Similarity). This also informs the decisions make when designing a conversion-optimized page. A strong understanding of theory will help create better hypotheses. All of the above knowledges is not required to know, but a firm grasp of psychological and emotional triggers that lead to sales will greatly help improve conversions.

Experience: All the case studies and theories in the world can’t make up for past experience. This is why CRO is often best left to experts; they can bring insights to the table that non-practitioners wouldn’t know. It’s not unusual to find that the most successful hypothesis is usually the one that springs from experience, not theory or case studies. A CRO expert who has already dealt with a site similar to in the same niche will likely have tons of ideas for improving conversion rates (Omniconvert, 2017).

3.2.4 Developing and Testing Page Variants

Developing test variants contain field changes on the form from the previous version. In addition, there is also the definition of the page name accompanied by the url that represents the page. The different versions of A, B, and C are determined by the difference in url sequence in the borrower’s data. The way traffic is distributed for each version so that the number of balanced visitors is also explained. Testing page variants contains an explanation of the implementation of Google Analytics on the entire page. The results of its implementation are seen in the dashboard of Google Analytics account and extracting data.

3.2.5 Analyzing Test Results

3.2.5.1 Bayessian Inference Algorithm

Let $X_1, ..., X_n$ be n observations sampled from a probability density $p(x|\theta)$. Write $p(x|\theta)$ if θ as a random variable and $p(x|\theta)$ represents the conditional probability density of X conditioned on θ. In contrast, write $p_\theta(x)$ if θ as a deterministic value.

Algoritma 3.1 Bayesian inference algorithm

1. Choose a probability density $\pi(\theta)$ — called the prior distribution — that expresses beliefs about a parameter θ.
2. Choose a statistical model $p(x|\theta)$ that reflects beliefs about x given θ.
3. After observing data $D_n = \{X_1, ..., X_n\}$, update beliefs and calculate the posterior distribution $p(\theta|D_n)$. The posterior distribution can be written as:

$$p(\theta|X_1, ..., X_n) = \frac{p(X_1, ..., X_n|\theta)\pi(\theta)}{p\left(X_1, ..., X_n\right)} = \frac{\mathcal{L}_n\left(\theta\right)\pi\left(\theta\right)}{c_n}\alpha\mathcal{L}_n\left(\theta\right)\pi\left(\theta\right)$$

where $\mathcal{L}_n\left(\theta\right) = \prod_{i=1}^{n} p\left(X_i|\theta\right)$ is the likelihood function and

$$c_n = p\left(X_1, ..., X_n\right) = \int p(X_1, ..., X_n|\theta)\pi(\theta)d\theta = \int \mathcal{L}_n\left(\theta\right)\pi\left(\theta\right)d\theta$$

is the normalizing constant, which is also called the evidedence (Liu and Wasserman, 2014).

3.2.5.2 Bernoulli Distribution

The Bernoulli distribution is a discrete distribution having two possible outcomes labelled by $n = 0$ and $n = 1$ in which $n = 1$ ("success") occurs with probability p and $n = 0$ ("failure") occurs with probability $q = 1 - p$, where $0 < p < 1$. It therefore has probability density function

$$p(n) = p^n(1-p)^{1-n}.$$

The distribution of heads and tails in coin tossing is an example of a Bernoulli distribution with $p = q = 1/2$. The Bernoulli distribution is the

simplest discrete distribution, and it the building block for other more complicated discrete distributions (Weisstein, 2018).

3.2.5.3 Pymc3

Pymc3 in analyzing test results of this research is used as probabilistic programming tools for bayesian inference method to build probabilistic models. Probabilistic programming offers an effective way to build complex models and allows us to focus more on model design, evaluation, and interpretation, and less on mathematical or computational details (Martin, 2016).

3.2.5.4 Metropolis-Hastings Algorithm

Metropolis-Hastings algorithm (MH) is one of special forms of the Hastings algorithm. The Hastings algorithm (HA) is a stochastic sampling technique widely used throughout computational science. As a Markov Chain Monte Carlo method, HA does not attempt to generate a sequence of independent samples from a "target distribution" $\pi(\cdot)$, defined on the state space(E, ε), but rather a Markov chain $\{Xn, n = 1, 2, 3, ...\}$ having $\pi(\cdot)$ as its invariant distribution. Although variates in the chain are not independent, they may nonetheless be used to estimate statistical expectations with respect to $\pi(\cdot)$.

Let U(0, 1) represent the uniform distribution on (0, 1). In order to use all subsequently described algorithms, given $X_n = x$, we require a "proposal density" $\gamma(\cdot|x)$ which may (or may not) depend on x, and whose variates can be generated by other means. Given $X_n = x \sim \pi(\cdot)$, we can generate $X_{n+1} \sim \pi(\cdot)$ by

Algoritma 3.2 Metropolis-Hastings algorithm

1. Generate $y \sim\gamma(\cdot|x)$ and $r \sim U$(0, 1)
2. If $r \leq \alpha_{HA}(x, y)$, output $X_{n+1} = y$
3. Else, output $X_{n+1} = x$

where densities are assumed to be symmetric (that is, $\gamma(x|y) = \gamma(y|x)$) and the acceptance probability in step 2 is $\alpha_{HA}(x, y) = min\left\{\frac{p(y)}{p(x)}, 1\right\}$ (Minh and

Minh, 2015).

Chapter 4

RESULTS AND DISCUSSION

4.1 Identifying Problems

PT Mekar or Sampoerna Wirausaha is a fintech company running a business with a peer-to-peer loan platform. In borrowing lending there are borrowers and lenders. Mekar Go is a website used to collect borrower data. The data collected on the current version of Mekar Go website shows the conversion of a borrower who completes data filling from start to finish less than 10%. This conversion percentage by the marketing team is too small and needs to be improved.

4.2 Defining Website Measurement

4.2.1 Business Objective

The desired business objective is to increase the number of borrowers that will be given a peer-to-peer lending loan.

4.2.2 Website Goal

The desired website goal is: increase the number of borrowers who register, increase the percentage completion of filling the registration form by the borrower.

4.2.3 Key Performance Metric

The metric used as the key performance metric is: the number of registrants per month and the percentage completion of form filling by the borrower upon filling in the registration form.

4.2.4 Target Metric

The expected target is 500 unique visitors who register for each version and percentage of data completion above 10% for each version. Both targets are tested within a month.

4.3 Developing a Hypothesis

The completion percentage of the registration form filling is influenced by the order of form filling. When changed will increase the percentage. The proposed solution is to test A / B testing against two new versions along with simplification and increment of data filling. The initial assumption, one of the two new versions will be better than the previous version.

4.4 Developing and Testing Page Variants

4.4.1 Developing Page Variants

The field that changes the data from the previous version in the table 4.1. The data changes section is categorized into two: personal details information and warranties. The type of change made is the removal of data for data that is considered unneeded. Optional is the part changed to not required, but the field still exists. The latter is new for new fields that do not exist in previous versions. This field change applies to all versions of A, B, and C.

Changes made to personal details information are to remove the business address field or place of employment, field of business type or type of work, length of business or duration of employment, number of employees, photos of place of business, employee status. Changes to fields that are made field optional on the personal details information is email field and photo ID card. The new field added to the personal details information section is the birth

date field. Then, the security section is only made field changes to be optional on the field of land and building photos and photos of motor vehicles.

Table 4.1: Data fields changes

Part	Fields Name	Remove	Optional	New
Personal Detail Information	Alamat Usaha / Tempat Bekerja (Provinsi, Kota, Kecamatan, Kelurahan)	v		
	Bidang Jenis Usaha / Jenis Pekerjaan	v		
	Lama Usaha / Lama Bekerja	v		
	Jumlah Karyawan	v		
	Foto Tempat Usaha	v		
	Status Karyawan	v		
	Email		v	
	Foto KTP		v	
	Tanggal Lahir			v
Jaminan	Foto Tanah + Bangunan		v	
	Foto Kendaraan Bermotor		v	

The data flow flows for versions A, B, and C are defined in the table 4.2. Each page is also considered a step or step and is named for each step. Step fill of all versions of data consists of: survey needs page, detail needs page, personal information page, detail loan access page, extra pesonal detail information page, and congrats page. Version B and C there is no extra step pesonal detail information page because put together in step pesonal detail information page. The url for all the steps is almost identical, the distinguish is the alphabetical a or b or c after the / ukm / on the url definition. This distinction of suffix makes it easy to differentiate traffic on the Google Analytics dashboard.

Table 4.2: Url definition

Step	Site A (/ukm/a/)	Site B (/ukm/b/)	Site C (/ukm/c/)
Survey Needs Page	/ukm/a/survei	/ukm/b/survei	/ukm/c/survei
Detail Needs Page	/ukm/a/detil-survei	/ukm/b/detil-survei	/ukm/c/detil-survei
Personal Information Page	/ukm/a/data-diri	/ukm/b/data-diri	/ukm/c/data-diri
Thanks Page	/ukm/a/terima-kasih	/ukm/b/terima-kasih	/ukm/c/terima-kasih
Detail Loan Access Page	/ukm/a/ pinjaman	/ukm/b/ pinjaman	/ukm/c/ pinjaman
Ekstra Pesonal Detail Information Page	/ukm/a/ tambahan-data-diri	-	-
Congrats Page	/ukm/a/selamat	/ukm/b/selamat	/ukm/c/selamat

The differentiator from A / B testing is the flow of data from all versions. Version A sequence of data filling is the same as the previous version. Meanwhile, B and C versions are new versions that will be compared which results are better. The sequence of data filling flow for each version is as follows:

- Data completion flow of version A: survey needs page - detail needs page - personal information page - thanks page - detail loan access page - ekstra pesonal detail information page - congrats page.
- Data completion flow of version B: personal information page - detail loan access page - thanks page - survey needs page - detail needs page - congrats page.
- Data completion flow of version C: detail loan access page - personal information page - thanks page - survey needs page - detail needs page - congrats page.

Traffic sharing for balanced versions A, B, and C is done on the server side. The programming language of the server part used is the python programming language. Syntax implementation of traffic sharing is in listing 4.1. The logic being implemented is: the first visitor will be directed to version A. Then, for the next visitor, query to the database to find out what version was last visited. If version A is the last visited, then the next visitor will be directed to version B. If version B is the last one visited, then the next visitor will be redirected to version C. If the C version is the last one visited, then the next visitors will be directed to version A.

Listing 4.1: Traffic splitting

```
class RoundRobin(models.Model):
    flow_type = models.CharField(max_length=1, null=True)
    partner_slug = models.SlugField(null=True)
    def next(self):
        latest = self.flow_type
        if latest == 'a':
            return 'b'
        elif latest == 'b':
            return 'c'
        elif latest == 'c':
            return 'a'

if RoundRobin.objects.count() == 0:
    next_flow = 'a'
else:
    latest = RoundRobin.objects.last()
    next_flow = latest.next()
```

4.4.2 Testing Page Variants

Testing page variant is done with the help of Google Analytics. The Google Analytics script applies to each page step for all versions. The Google Analytic Syntax script used in the listing listings 4.2. PIn syntax "gtag ('config', 'UA-70057602-1');" is a tracking marker owned by Mekar GO. Unique code UA-70057602-1 is obtained from Google Analytic's dashboard on PT Mekar's Google account.

Listing 4.2: Google Analytics implementation

```
<!-- Global Site Tag (gtag.js) - Google Analytics -->
<script async src="https://www.googletagmanager.com/gtag/
    js?id=UA-70057602-1">
</script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());
  gtag('config', 'UA-70057602-1');
</script>
```

Google Analytic Tracking is conducted from September 24, 2017 to October 23, 2017. High traffic occurs on September 3, 2017, October 9-10, 2017, October 13, 2017 and October 16, 2017. Graph fluctuations of visitors traffic Mekar Go Website website in figure4.1 obtained from the dashboard of PT Mekar's Google Analytics account. The traffic shown is the combined traffic for all versions A, B, and C.

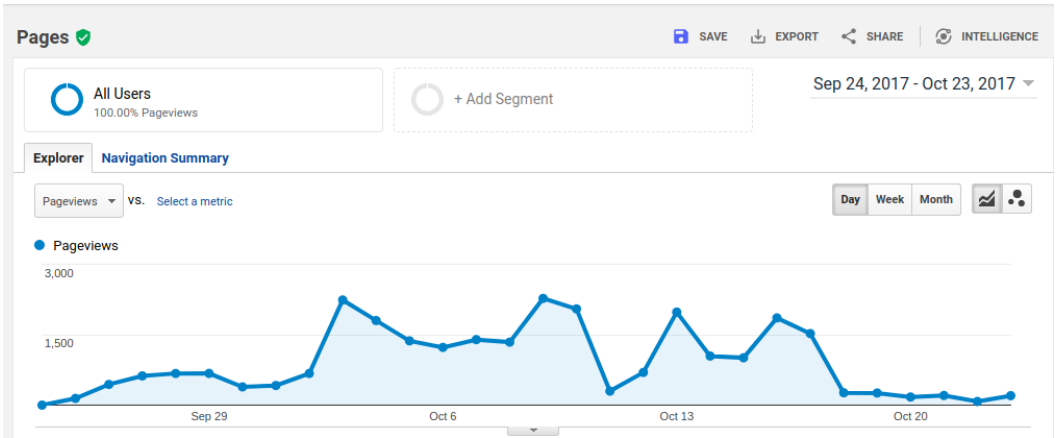
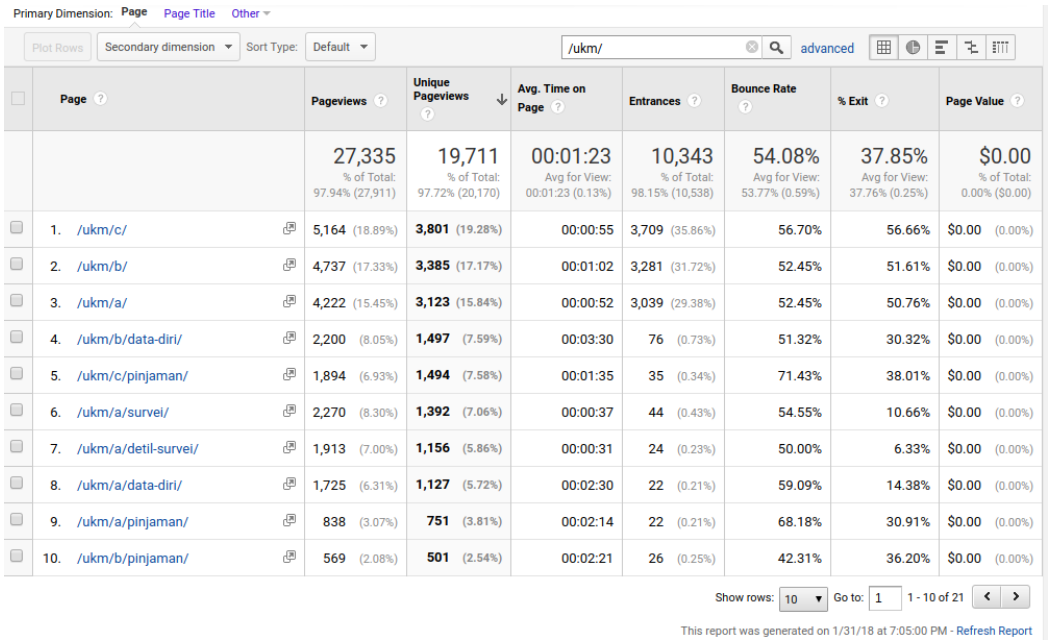


Figure 4.1: Graphic google analytics mekargo.id

Below the traffic graph there is a detailed table of visitor statistics on each page. The visitor statistics table on the Google Analytic dashboard looks like in figure4.2. Not all data contained in the table is used for A / B testing process analysis. The row used from the statistics table of pengunujung is row unique pageviews. This is to match the target metrics that measure unique visitors. Figure 4.2 shows that three initial data of unique visitor on version A, B and C have fulfilled the target of unique visitor greater than 500.



Primary Dimension: Page | Page Title | Other

Plot Rows | Secondary dimension | Sort Type: Default | /ukm/ | advanced

	Page	Pageviews	Unique Pageviews	Avg. Time on Page	Entrances	Bounce Rate	% Exit	Page Value
		27,335 % of Total: 97.94% (27,911)	19,711 % of Total: 97.72% (20,170)	00:01:23 Avg for View: 00:01:23 (0.13%)	10,343 % of Total: 98.15% (10,538)	54.08% Avg for View: 53.77% (0.59%)	37.85% Avg for View: 37.76% (0.25%)	$0.00 % of Total: 0.00% ($0.00)
1.	/ukm/c/	5,164 (18.89%)	3,801 (19.28%)	00:00:55	3,709 (35.86%)	56.70%	56.66%	$0.00 (0.00%)
2.	/ukm/b/	4,737 (17.33%)	3,385 (17.17%)	00:01:02	3,281 (31.72%)	52.45%	51.61%	$0.00 (0.00%)
3.	/ukm/a/	4,222 (15.45%)	3,123 (15.84%)	00:00:52	3,039 (29.38%)	52.45%	50.76%	$0.00 (0.00%)
4.	/ukm/b/data-diri/	2,200 (8.05%)	1,497 (7.59%)	00:03:30	76 (0.73%)	51.32%	30.32%	$0.00 (0.00%)
5.	/ukm/c/pinjaman/	1,894 (6.93%)	1,494 (7.58%)	00:01:35	35 (0.34%)	71.43%	38.01%	$0.00 (0.00%)
6.	/ukm/a/survei/	2,270 (8.30%)	1,392 (7.06%)	00:00:37	44 (0.43%)	54.55%	10.66%	$0.00 (0.00%)
7.	/ukm/a/detil-survei/	1,913 (7.00%)	1,156 (5.86%)	00:00:31	24 (0.23%)	50.00%	6.33%	$0.00 (0.00%)
8.	/ukm/a/data-diri/	1,725 (6.31%)	1,127 (5.72%)	00:02:30	22 (0.21%)	59.09%	14.38%	$0.00 (0.00%)
9.	/ukm/a/pinjaman/	838 (3.07%)	751 (3.81%)	00:02:14	22 (0.21%)	68.18%	30.91%	$0.00 (0.00%)
10.	/ukm/b/pinjaman/	569 (2.08%)	501 (2.54%)	00:02:21	26 (0.25%)	42.31%	36.20%	$0.00 (0.00%)

Show rows: 10 | Go to: 1 | 1 - 10 of 21

This report was generated on 1/31/18 at 7:05:00 PM - Refresh Report

Figure 4.2: Table google analytics mekargo.id

4.5 Analyzing Test Results

Data obtained from google analytics is done processing and analysis with bayesian inference method with pymc3.

```
In [1]: import pymc3 as pm
        import numpy as np
        %matplotlib inline
        from IPython.core.pylabtools import figsize
        import matplotlib.pyplot as plt
        import scipy.stats as stats
        figsize(12.5, 4)
```

Figure 4.3: Import python library

he first syntax to analyze A / B test results in this study is to import all required python libraries ie pymc3, numpy, Ipython, matplotlib, and scipy. This syntax is written in the notebook jupyter environment. The syntax is in the figure 4.3.

```
In [2]: true_A = 120
        true_B = 129
        true_C = 84

        N_sample = 1156
```

Figure 4.4: Initializing sample

Figure 4.4 contains initialization syntax initial value true_A, true_B, true_C, and N_sample. Initial value true_A, true_B, and true_C function as a storage variable value of the number of users completing the data completion process on versions A, B, and C. Meanwhile, N_sample as the storage variable of the number of samples used. Initial value true_A as many as 120 users who completed the process of charging data on version A, the initial value true_B as many as 129 users who completed the process of data filling in version B, and the initial value true_C as many as 84 users who completed the process of data filling in version C. Meanwhile, N_sample 1156 users as samples that perform the process of filling data on each version A, B, and C. Sample used the same value for all versions.

```
In [3]: true_p_A = true_A/float(N_sample)
        true_p_B = true_B/float(N_sample)
        true_p_C = true_C/float(N_sample)

        print("true p_A:", true_p_A)
        print("true p_B:", true_p_B)
        print("true p_C:", true_p_C)

true p_A: 0.10380622837370242
true p_B: 0.1115916955017301
true p_C: 0.0726643598615917
```

Figure 4.5: Calculating probability

The syntax of the calculation operation found in figure4.5 is the first step in bayesian inference algorithm 3.1. The result of this syntax operation determines the probability density value or the so-called prior distribution describing beliefs about a parameter. In this case the number of users completing the charging data compared to the total number of users who do the data

(which completed plus the not). Variable true_p_A is prior distribution for version A, variable true_p_B is prior distribution for version B, and variable true_p_C is prior distribution for version C. The priority distribution for each version is as follows:

true_p_A = 0.10380622837370242

true_p_B = 0.1115916955017301

true_p_C = 0.0726643598615917

The values true_p_A, true_p_B, and true_p_C are obtained by performing a calculation operation against the initial initiation values gained previously. This value indicates the completion percentage of completing the completed data for each version. The variable true_p_A indicates the completion percentage of A version of data charging as much as 10.38%. The variable true_p_B indicates the completion percentage of data charging A version of 11.16%. The variable true_p_C shows the completion percentage of A version of data filling as much as 7.27%. Versions A and B have met the metric targets that want a percentage completion of data charging above 10 percent.

```
In [4]: observations_A = stats.bernoulli.rvs(true_p_A, size=N_sample)
        observations_B = stats.bernoulli.rvs(true_p_B, size=N_sample)
        observations_C = stats.bernoulli.rvs(true_p_C, size=N_sample)

        print("Obs from Site A: ", observations_A[:30], "...")
        print("Obs from Site B: ", observations_B[:30], "...")
        print("Obs from Site C: ", observations_C[:30], "...")

        Obs from Site A:  [0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0] ...
        Obs from Site B:  [0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0] ...
        Obs from Site C:  [0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0] ...
```

Figure 4.6: Bernoulli distribution observation

This stage is the second step in the Bayesian inference algorithm 3.1 that is selecting the statistical model that reflects beliefs with the prior distribution selected in the previous step. This study chose statistical model of bernoulli distribution to reflect belief on prior distribution in the previous step. Implementation on python syntax using stats method imported from library scipy. The result of the bernoulli distribution is the numpy array containing the random numbers 0 and 1 generated according to the prior distribution for each version A, B, and C. The number 1 represents true and the number 0 represents the false value. The value of the bernoulli distribution generated from scipy is affected by the input of true_p_A, true_p_B,

true_p_C and N_sample. The syntax of statistical modeling of the bernoulli distribution and the result can be seen in figure 4.6. The bernoulli distribution value for each version is stored in observation_A, observation_B, and observation_C variables.

```
In [5]: print(np.mean(observations_A))
        print(np.mean(observations_B))
        print(np.mean(observations_C))

        print(np.sum(observations_A))
        print(np.sum(observations_B))
        print(np.sum(observations_C))

        0.0951557093426
        0.117647058824
        0.0726643598616
        110
        136
        84
```

Figure 4.7: Sum and mean of bernoulli distribution observation

The average value of the observations of bernoulli is obtained with the help of the numpy library with the mean method. The values are:

mean_p_A = 0.098615916955

mean_p_B = 0.11937716263

mean_p_C = 0.0743944636678

The average value of these noulli observations becomes an update of belief values in the third step of the bayessian inference algorithm 3.1. The average value of noulli observations is called true frequency. The true frequency value in this case is the actual opportunity value of the user completing the data filling until the end is true. The observation value of noulli shows how much true value is generated on the distribution. The true value of each version is as follows: version A is represented as sum_true_p_A variable 114 times true, version B represented sum_true_p_B variable as 138 times true, and C version represented sum_true_p_C variable as much as 86 times true. After that, perform posterior distribution calculations.

```
In [6]: with pm.Model() as model:
            p_A = pm.Uniform("p_A", 0, 1)
            p_B = pm.Uniform("p_B", 0, 1)
            p_C = pm.Uniform("p_C", 0, 1)

            # Define the deterministic delta function. This is our unknown of interest.
            delta_A_B = pm.Deterministic("delta_A_B", p_A - p_B)
            delta_A_C = pm.Deterministic("delta_A_C", p_A - p_C)
            delta_B_C = pm.Deterministic("delta_B_C", p_B - p_C)

            # Set of observations, in this case we have three observation datasets.
            obs_A = pm.Bernoulli("obs_A", p_A, observed=observations_A)
            obs_B = pm.Bernoulli("obs_B", p_B, observed=observations_B)
            obs_C = pm.Bernoulli("obs_C", p_C, observed=observations_C)

            step = pm.Metropolis()
            trace = pm.sample(20000, step=step)
            burned_trace=trace[1000:]
        100%|| 20500/20500 [00:31<00:00, 643.38it/s]
```

Figure 4.8: Pymc probabilistic model

Posterior distribution calculation with MH algorithm 3.2 on pymc probability models. The syntax is found in figure 4.8. Syntax in this section does not reflect coherently according to the MH algorithm 3.2. he displayed syntax only provides the value of any input required with the use of the probability models of pymc used. The algorithm process is performed behind the scenes by the probability models of pymc. Thus, the Metropolis-Hastings algorithm is executed behind the scenes by the method of pm.Metropolis (). The sample of the Metropolis-Hastings algorithm used is 2000. The meaning of trace [1000:] means that sampling values are taken from 1000 to last, this is to take 95% sample and ignore 0.05 (1000 divided by 20000) convergent.

```
In [7]: p_A_samples = burned_trace["p_A"]
        p_B_samples = burned_trace["p_B"]
        p_C_samples = burned_trace["p_C"]
        delta_A_B_samples = burned_trace["delta_A_B"]
        delta_A_C_samples = burned_trace["delta_A_C"]
        delta_B_C_samples = burned_trace["delta_B_C"]
```

Figure 4.9: Pymc probabilistic model results

The output of figure 4.8 is the posterior distribution result required for the analysis process. The output is extracted into several variables, as in figure 4.9. The variable p_A_samples is the posterior end value of the distribution of version A. The variable p_B_samples is the posterior end value of

the distribution of version B. The p_C_samples variable is the final value of the posterior distribution version C. The variable delta_A_B_samples is the posterior end value of the comparison distribution between version A and version B. The delta_A_C_samples variable is the final value posterior distribution comparison between version A and version C. The delta_B_C_samples variable is the final value of the posterior distribution comparison between version B and version C.

```
In [8]: figsize(12.5, 10)

        #histogram of posteriors

        ax = plt.subplot(311)

        plt.xlim(.06, .2)
        plt.hist(p_A_samples, histtype='stepfilled', bins=25, alpha=0.85,
                 label="posterior of $p_A$", color="#A60628", normed=True)
        plt.vlines(true_p_A, 0, 80, linestyle="--", label="true $p_A$ (unknown)")
        plt.legend(loc="upper right")
        plt.title("Posterior distributions of $p_A$, $p_B$, and delta unknowns")

Out[8]: Text(0.5,1,'Posterior distributions of $p A$, $p B$, and delta unknowns')
```

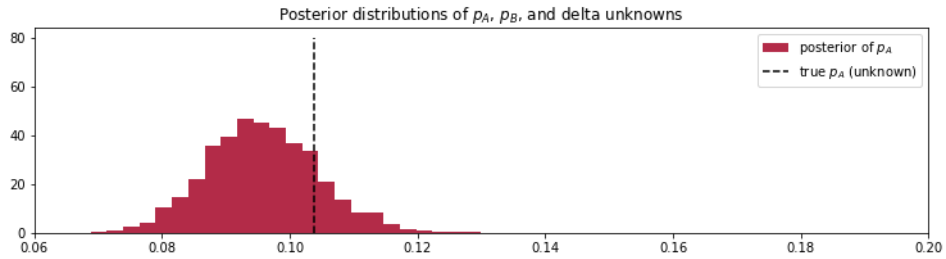


Figure 4.10: Posteriror distribution P_A

Figure 4.10 is a visualization of posterior distribution for version A. Axis x is plotted in the range of 0.06 to 0.2, but the posterior value distribution for version A lies between 0.07 and 0.13. The dotted vertical line is the location of the prior distribution value of A on the variable true_p_A = 0.10380622837370242 that has been declared at the beginning. The most posterior value is at about 0.092.

```
In [9]: ax = plt.subplot(312)

        plt.xlim(.06, .2)
        plt.hist(p_B_samples, histtype='stepfilled', bins=25, alpha=0.85,
                 label="posterior of $p_B$", color="#467821", normed=True)
        plt.vlines(true_p_B, 0, 80, linestyle="--", label="true $p_B$ (unknown)")
        plt.legend(loc="upper right")

Out[9]: <matplotlib.legend.Legend at 0x7f03334969e8>
```

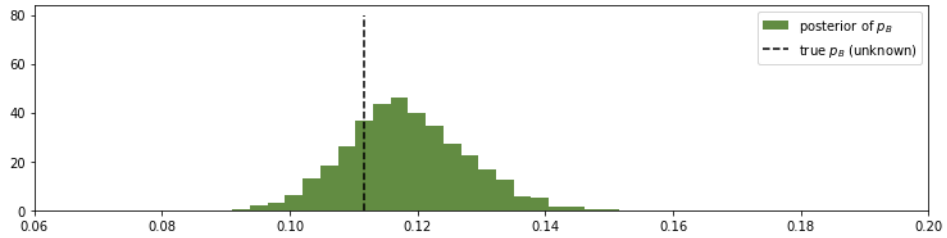


Figure 4.11: Posteriror distribution P_B

Figure 4.11 is a visualization of posterior distribution for version B. Axis x is plotted in the range of 0.06 to 0.2. However, the posterior value distribution for version B lies between 0.09 and 0.15. The dotted vertical line is the location of the prior distribution value of B on the variable true_p_B = 0.1115916955017301 which has been declared at the beginning. The most posterior value is present at about 0.118.

```
In [10]: ax = plt.subplot(313)

         plt.xlim(0, .1)
         plt.hist(p_C_samples, histtype='stepfilled', bins=25, alpha=0.85,
                  label="posterior of $p_C$", color="#D6F841", normed=True)
         plt.vlines(true_p_C, 0, 80, linestyle="--", label="true $p_C$ (unknown)")
         plt.legend(loc="upper right")

Out[10]: <matplotlib.legend.Legend at 0x7f0332445a90>
```

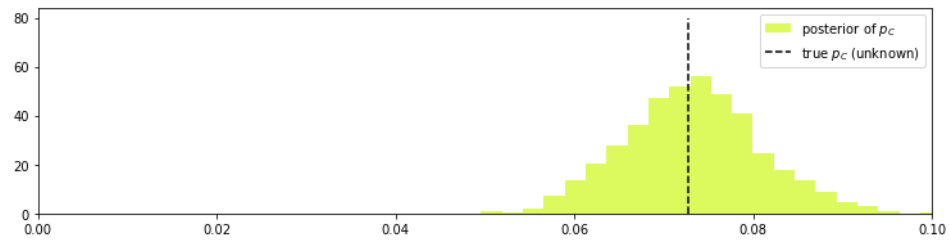


Figure 4.12: Posteriror distribution P_C

Figure 4.12 is a visualization of posterior distribution for version C. Axis x is plotted in the range of 0.00 to 0.1. However, the posterior value distribution for version C lies between 0.05 and 0.09. The dashed vertical line is the location of the prior distribution value of C on the variable true_p_C = 0.0726643598615917 which has been declared at the beginning. The most

posterior value is present at about 0.072.

Table 4.3: List of variables of A,B, and C version

Variable name	Version A	Version B	Version C
users completing the data completion	true_A 120	true_B 129	true_C 84
completion percentage	true_p_A 0.10380622837370242	true_p_B 0.1115916955017301	true_p_C 0.0726643598615917
average of the bernoulli observations	sum_true_p_A 114	sum_true_p_B 138	sum_true_p_C 86
mean of the bernoulli observations	mean_p_A 0.098615916955	mean_p_B 0.11937716263	mean_p_C 0.0743944636678

Table 4.3 shows the comparison of variables in versions A, B, and C. The variables which compared are about users completing the data completion, completion percentage, average of the bernoulli observations and mean of the bernoulli observations.

The next few graphs(figure 4.13, 4.14, and 4.15) are the posterior visualization of the combination combinations of all versions. Combination combinations are performed for each of the two versions. The resulting comparison combination is the comparison between versions A and B called delta A-B, the comparison between versions A and C is also called A-C delta, and the comparison between versions B and C is also called B-C delta.

```
In [11]: ax = plt.subplot(313)
         plt.hist(delta_A_B_samples, histtype='stepfilled', bins=30, alpha=0.85,
                  label="posterior of delta A-B", color="#7A68A6", normed=True)
         plt.vlines(true_p_A - true_p_B, 0, 60, linestyle="--",
                    label="true delta (unknown)")
         plt.vlines(0, 0, 60, color="black", alpha=0.2)
         plt.legend(loc="upper right");
```

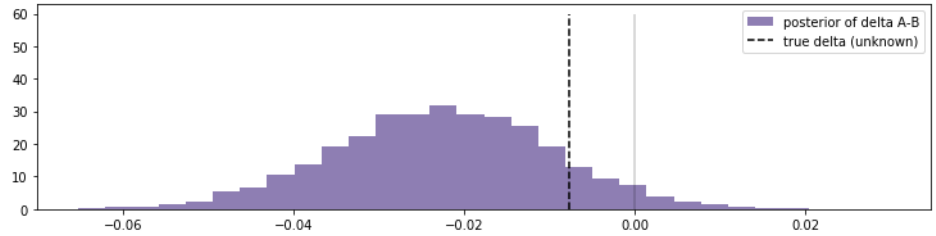


Figure 4.13: Posteriror distribution delta A-B

Figure 4.13 is a visualization of probabilistic distribution for delta A-B. The posterior value distribution for the A-B delta is between -0.05 to 0.02. Dotted vertical line is where the prior distribution value of delta A-B is at a value of about -0.01. The unbroken vertical line shows the value 0 in axis x. Posterior distribution of distribution is more on the left value of 0 or on a negative value. That is, the subtracting variable from the A-B delta comparison becomes the better version. Version B becomes a better version than version A because version B as a delimiter on the A-B delta.

```
In [12]: ax = plt.subplot(313)
         plt.hist(delta_A_C_samples, histtype='stepfilled', bins=30, alpha=0.85,
                  label="posterior of delta A-C", color="#B868B6", normed=True)
         plt.vlines(true_p_A - true_p_C, 0, 60, linestyle="--",
                    label="true delta (unknown)")
         plt.vlines(0, 0, 60, color="black")
         plt.legend(loc="upper right");
```

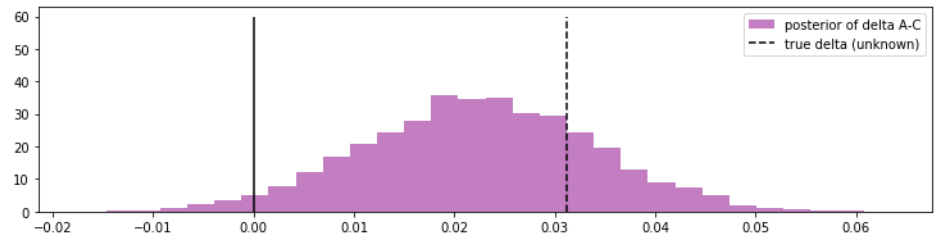


Figure 4.14: Posteriror distribution delta A-C

Figure 4.14 is a visualization of probabilistic distribution for delta A-B. The posterior value distribution for the A-C delta is between -0.015 to 0.06. Dotted vertical line is where the prior distribution value of delta A-C is at

value around 0.031. The unbroken vertical line shows the value 0 in axis x. Posterior distribution of distribution is more to the right of value 0 or to a positive value. That is, the reduced variables from the A-C delta ratio to the better versions. Version A becomes a better version than version C because version A is reduced on the A-C delta.

```
In [13]: ax = plt.subplot(313)
         plt.hist(delta_B_C_samples, histtype='stepfilled', bins=30, alpha=0.85,
                  label="posterior of delta B-C", color="#24B8A6", normed=True)
         plt.vlines(true_p_B - true_p_C, 0, 60, linestyle="--",
                    label="true delta (unknown)")
         plt.vlines(0, 0, 60, color="black")
         plt.legend(loc="upper right");
```

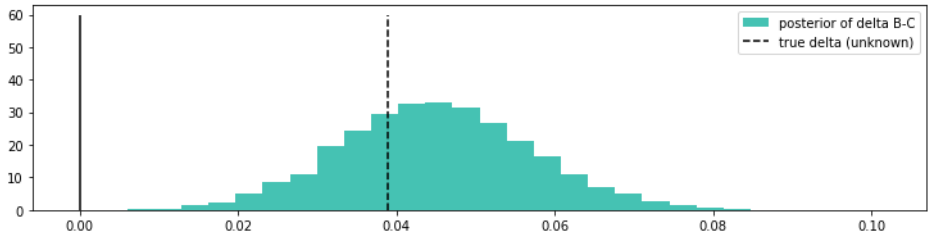


Figure 4.15: Posteriror distribution delta B-C

Figure 4.13 is a visualization of probabilistic distribution for delta A-B. The posterior value distribution for the A-B delta is between 0.01 and 0.082. Dotted vertical line is where the prior distribution value of delta A-B is at a value of about 0.038. The solid vertical line shows the value 0 in axis x. The posterior distribution of the distribution is all to the right of value 0 or to a positive value. That is, the reduced variables of the B-C delta ratio become better versions. Version B is a better version than version C because version B is reduced on the B-C delta.

```
In [14]: # Count the number of samples less than 0, i.e. the area under the curve
         # before 0, represent the probability that site A is worse than site B.
         print("Probability site A is WORSE than site B: %.3f" % \
             np.mean(delta_A_B_samples < 0))

         print("Probability site A is BETTER than site B: %.3f" % \
             np.mean(delta_A_B_samples > 0))

         print("\nProbability site A is WORSE than site C: %.3f" % \
             np.mean(delta_A_C_samples < 0))

         print("Probability site A is BETTER than site C: %.3f \n" % \
             np.mean(delta_A_C_samples > 0))

         print("Probability site B is WORSE than site C: %.3f" % \
             np.mean(delta_B_C_samples < 0))

         print("Probability site B is BETTER than site C: %.3f" % \
             np.mean(delta_B_C_samples > 0))

         Probability site A is WORSE than site B: 0.961
         Probability site A is BETTER than site B: 0.039

         Probability site A is WORSE than site C: 0.025
         Probability site A is BETTER than site C: 0.975

         Probability site B is WORSE than site C: 0.000
         Probability site B is BETTER than site C: 1.000
```

Figure 4.16: Final result combination of all variants

The result of postireior distribution comparison analysis has been conducted, disclose how the rankings of all versions. Figure 4.16 shows the value of the percentage of confidence for all combinations of comparisons. Version A compared to version B the winner is version B, with confidence version A worse 96.1 percent (0.961) percent of version B. Version A compared to version C the winner is version A, with confidence version A better 97.5 percent (0.975) than version C. Version B compared to version C the winner is version B, with confidence version B is better 100 percent (1.00) of version C. So the best version is version B because version B > version A > C version.

Chapter 5

CONCLUSION AND SUGGESTIONS

5.1 Conclusion

After performing A / B testing on the website mekargo.id with stages of research methods conducted include: identifying problems, defining website measurement, developing a hypothesis, developing and testing page variants, and analyzing test results. And analyzing the results of A / B testing with bayesian inference and pymc3 tools is done through Metropolis-Hastings algorithm as sampling technique, ipython and jupyter notebook as programming environment, scipy to produce bernoulli distribution, numpy as numerical data storage in array form, and matplotlib as viewer graph of result of data processing. It can be concluded that the final result of all combinations of probabilities generated by version B is the best result because it is always better when compared with versions A and C.

5.2 Suggestions

Subsequent research can be attempted to perform A / B testing with more versions and development of bayesian inference method with multi-arm bandit algorithm.

Bibliography

Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A., Belopolsky, A., et al. (2016). Theano: A python framework for fast computation of mathematical expressions. *arXiv preprint*.

CXL (2017). *How to Build a Strong A/B Testing Plan That Gets Results*. [Online]. Available at: https://conversionxl.com/blog/how-to-build-a-strong-ab-testing-plan-that-gets-results/ [Accessed 17 February 2018].

Davidson-Pilon, C. (2015). *Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference*. Addison-Wesley Data & Analytics Series. Pearson Education.

Dixon, E., Enos, E., and Brodmerkle, S. (2015). A/b testing. US Patent 9,077,780.

Fandango, A. and Idris, I. (2017). *Python Data Analysis - Second Edition*. Packt Publishing.

Fernández-i Marín, X. et al. (2016). ggmcmc: Analysis of mcmc samples and bayesian inference. *Journal of Statistical Software*, 70(9).

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2014). *Bayesian data analysis*, volume 2. CRC press Boca Raton, FL.

Goldberg, D. and Johndrow, J. E. (2017). A decision theoretic approach to a/b testing. *arXiv preprint arXiv:1710.03410*.

Hunter, J., Dale, D., Firing, E., and Droettboom, M. (2014). The matplotlib development team. matplotlib: Python plotting – documentation. 2013.

Jiménez, J. and Ginebra, J. (2017). pygpgo: Bayesian optimization for python. *The Journal of Open Source Software*.

Johari, R., Pekelis, L., and Walsh, D. J. (2015). Always valid inference: Bringing sequential analysis to a/b testing. *arXiv preprint arXiv:1512.04922*.

Jones, E., Oliphant, T., and Peterson, P. (2014). {SciPy}: open source scientific tools for {Python}.

Kaufmann, E., Cappé, O., and Garivier, A. (2014). On the complexity of a/b testing. In *Conference on Learning Theory*, pages 461–481.

Kelsey, T. (2017). *Introduction to Google Analytics: A Guide for Absolute Beginners*. Apress.

Kemenkeu (2015). *Peran Penting UKM Dorong Perekonomian Indonesia*. [Online]. Available at: https://www.kemenkeu.go.id/Berita/peran-penting-ukm-dorong-perekonomian-indonesia [Accessed 17 January 2018].

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J. B., Grout, J., Corlay, S., et al. (2016). Jupyter notebooks-a publishing format for reproducible computational workflows. In *ELPUB*, pages 87–90.

Kohavi, R., Deng, A., Frasca, B., Walker, T., Xu, Y., and Pohlmann, N. (2013). Online controlled experiments at large scale. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1168–1176. ACM.

Kohavi, R. and Longbotham, R. (2017). Online controlled experiments and a/b testing. In *Encyclopedia of Machine Learning and Data Mining*, pages 922–929. Springer.

Liu, H. and Wasserman, L. (2014). *Statistical Machine Learning Chapter 12 Bayesian Inference*. [Online]. Available at: http://www.stat.cmu.edu/ larry/=sml/Bayes.pdf [Accessed 17 January 2018].

Lubanovic, B. (2014). *Introducing Python: Modern Computing in Simple Packages*. O'Reilly Media.

Lutz, M. (2013). *Learning Python: Powerful Object-Oriented Programming*. Safari Books Online. O'Reilly Media.

Martin, O. (2016). *Bayesian Analysis with Python*. Packt Publishing.

Mekar (2017). *Mekar FAQ*. [Online]. Available at: https://mekar.id/faq/ [Accessed 17 January 2018].

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., et al. (2017). Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103.

Minh, D. D. and Minh, D. L. (2015). Understanding the hastings algorithm. *Communications in Statistics-Simulation and Computation*, 44(2):332–349.

Omniconvert (2017). *How to create an A/B test*. [Online]. Available at: https://www.omniconvert.com/how-to-create-an-a-b-test [Accessed 17 January 2018].

Pourzanjani, A. A., Jiang, R. M., Atzberger, P. J., and Petzold, L. R. (2017). General bayesian inference over the stiefel manifold via the givens transform. *arXiv preprint arXiv:1710.09443*.

Ragan-Kelley, M., Perez, F., Granger, B., Kluyver, T., Ivanov, P., Frederic, J., and Bussonnier, M. (2014). The jupyter/ipython architecture: a unified view of computational research, from interactive exploration to communication and publication. In *AGU Fall Meeting Abstracts*.

Rougier, N. P., Hinsen, K., Alexandre, F., Arildsen, T., Barba, L. A., Benureau, F. C., Brown, C. T., De Buyl, P., Caglayan, O., Davison, A. P., et al. (2017). Sustainable computational science: the rescience initiative. *PeerJ Computer Science*, 3:e142.

Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. (2016). Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55.

Smith, A. M., Niemeyer, K. E., Katz, D. S., Barba, L. A., Githinji, G., Gymrek, M., Huff, K. D., Madan, C. R., Mayes, A. C., Moerman, K. M., et al. (2018). Journal of open source software (joss): design and first-year review. *PeerJ Computer Science*, 4:e147.

Strickland, C. M., Burdett, R. L., Denham, R., and Mengersen, K. L. (2014). Pyssm: a python module for bayesian inference of linear gaussian state space models. *Journal of Statistical Software*, 57(6).

Taylor, B., Davies, T., Rowlingson, B., and Diggle, P. (2015). Bayesian inference and data augmentation schemes for spatial, spatiotemporal and multivariate log-gaussian cox processes in r. *Journal of Statistical Software*, 63:1–48.

Van Der Walt, S., Colbert, S., and Varoquaux, G. (2016). The numpy array: a structure for efficient numerical computation (2011).

Vo, T., Czygan, M., et al. (2015). *Getting started with Python data analysis: learn to use powerful Python libraries for effective data processing and analysis*. Packt Publ.

Vohl, D., Barnes, D. G., Fluke, C. J., Poudel, G., Georgiou-Karistianis, N., Hassan, A. H., Benovitski, Y., Wong, T. H., Kaluza, O. L., Nguyen, T. D., et al. (2016). Large-scale comparative visualisation of sets of multidimensional data. *PeerJ Computer Science*, 2:e88.

Weber, J. (2015). *Practical Google Analytics and Google Tag Manager for Developers*. Apress.

Weisstein, E. W. (2018). *Bernoulli Distribution*. [Online]. Available at: http://mathworld.wolfram.com/BernoulliDistribution.html [Accessed 23 February 2018].

Wilson, G., Perez, F., and Norvig, P. (2014). Teaching computing with the ipython notebook. In *Proceedings of the 45th ACM technical symposium on Computer science education*, pages 740–740. ACM.

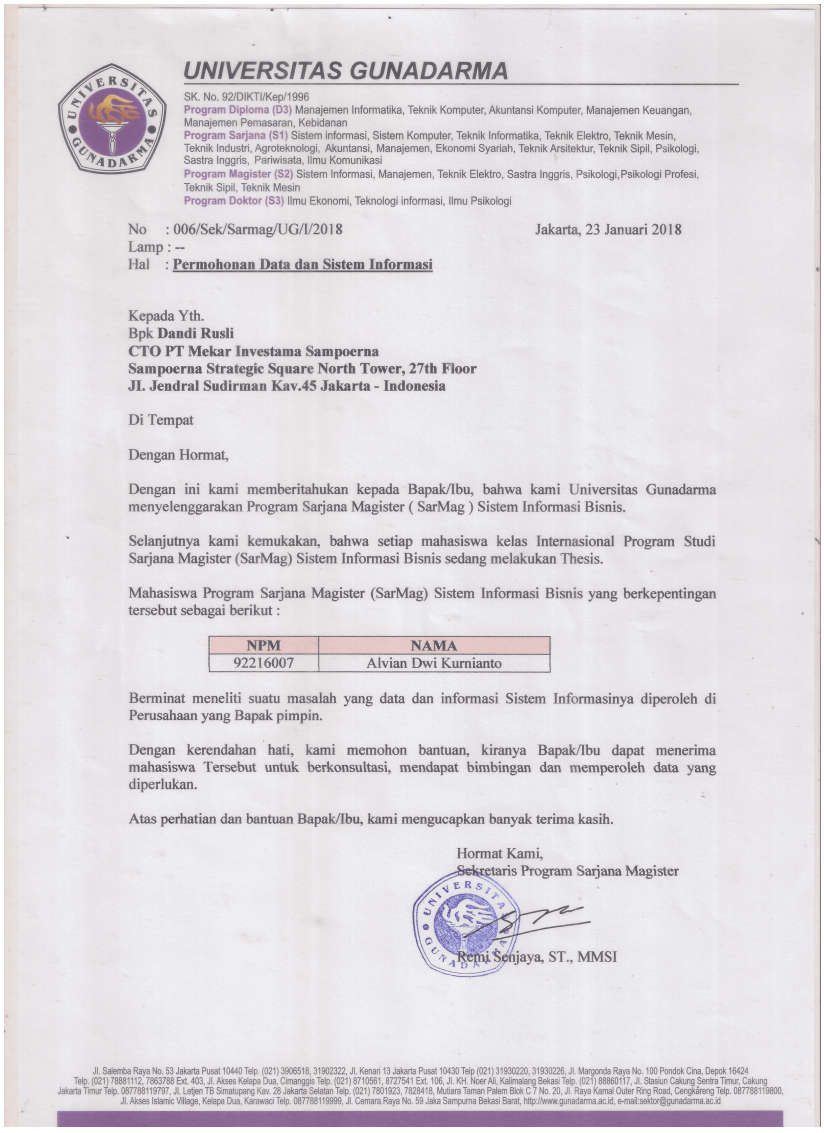
Wilson, J. D. and Uminsky, D. T. (2017). The power of a/b testing under interference. *arXiv preprint arXiv:1710.03855*.

Wuisan, N. Y. (2017). Pengaruh kredit bank terhadap pendapatan pelaku-pelaku umkm (khususnya meubel) di desa leilem, kec. sonder, kab. minahasa. *JURNAL BERKALA ILMIAH EFISIENSI*, 17(01).

Yang, L. and Perrin, J. M. (2014). Tutorials on google analytics: How to craft a web analytics report for a library web site. *Journal of Web Librarianship*, 8(4):404–417.

APPENDIX

Figure A.1: Letter of data and information permission request



UNIVERSITAS GUNADARMA

SK. No. 92/DIKTI/Kep/1996
Program Diploma (D3) Manajemen Informatika, Teknik Komputer, Akuntansi Komputer, Manajemen Keuangan, Manajemen Pemasaran, Kebidanan
Program Sarjana (S1) Sistem informasi, Sistem Komputer, Teknik Informatika, Teknik Elektro, Teknik Mesin, Teknik Industri, Agroteknologi, Akuntansi, Manajemen, Ekonomi Syariah, Teknik Arsitektur, Teknik Sipil, Psikologi, Sastra Inggris, Pariwisata, Ilmu Komunikasi
Program Magister (S2) Sistem Informasi, Manajemen, Teknik Elektro, Sastra Inggris, Psikologi,Psikologi Profesi, Teknik Sipil, Teknik Mesin
Program Doktor (S3) Ilmu Ekonomi, Teknologi informasi, Ilmu Psikologi

No : 006/Sek/Sarmag/UG/I/2018 Jakarta, 23 Januari 2018
Lamp : --
Hal : **Permohonan Data dan Sistem Informasi**

Kepada Yth.
Bpk **Dandi Rusli**
CTO PT Mekar Investama Sampoerna
Sampoerna Strategic Square North Tower, 27th Floor
Jl. Jendral Sudirman Kav.45 Jakarta - Indonesia

Di Tempat

Dengan Hormat,

Dengan ini kami memberitahukan kepada Bapak/Ibu, bahwa kami Universitas Gunadarma menyelenggarakan Program Sarjana Magister (SarMag) Sistem Informasi Bisnis.

Selanjutnya kami kemukakan, bahwa setiap mahasiswa kelas Internasional Program Studi Sarjana Magister (SarMag) Sistem Informasi Bisnis sedang melakukan Thesis.

Mahasiswa Program Sarjana Magister (SarMag) Sistem Informasi Bisnis yang berkepentingan tersebut sebagai berikut :

NPM	NAMA
92216007	Alvian Dwi Kurnianto

Berminat meneliti suatu masalah yang data dan informasi Sistem Informasinya diperoleh di Perusahaan yang Bapak pimpin.

Dengan kerendahan hati, kami memohon bantuan, kiranya Bapak/Ibu dapat menerima mahasiswa Tersebut untuk berkonsultasi, mendapat bimbingan dan memperoleh data yang diperlukan.

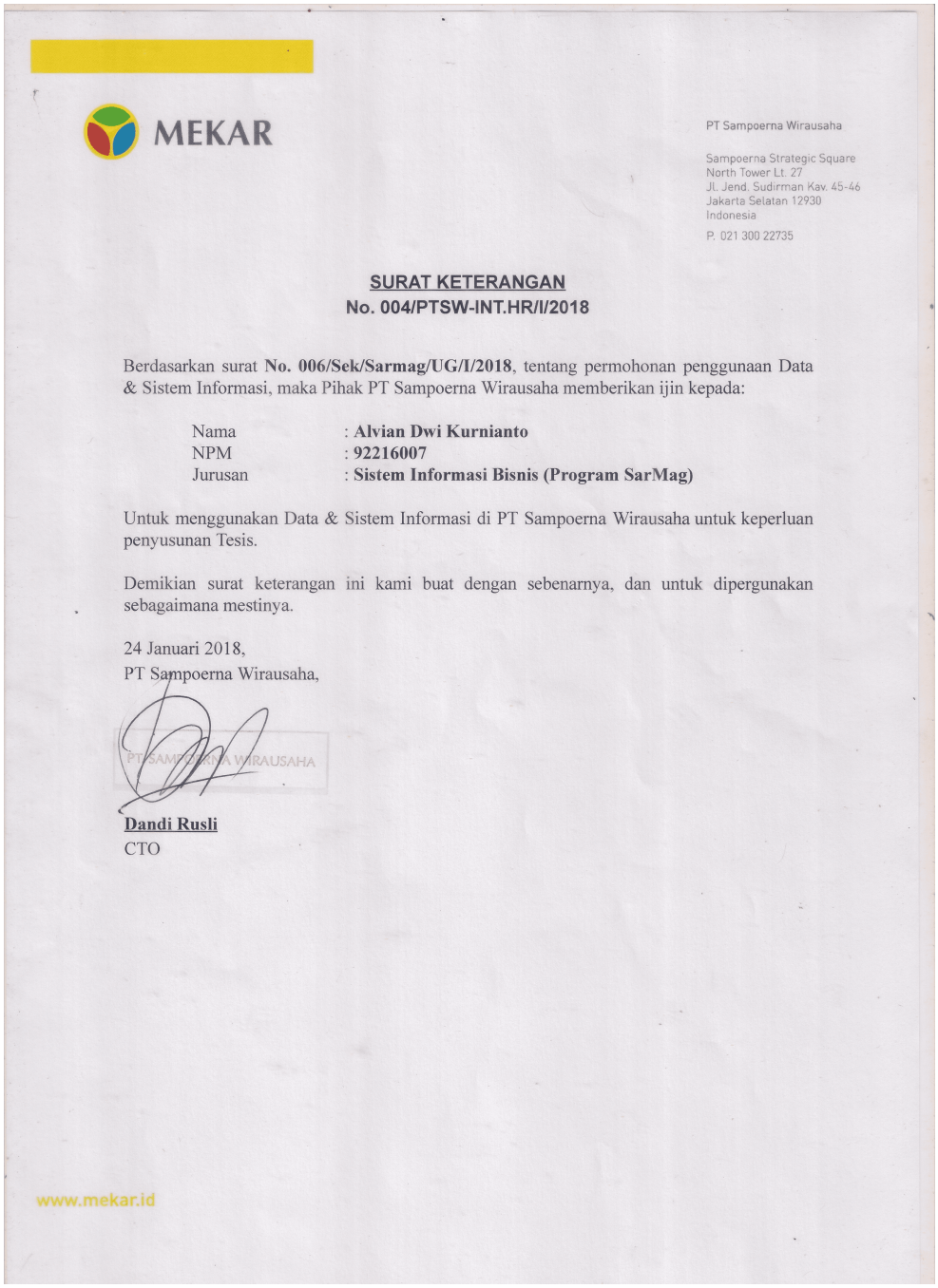
Atas perhatian dan bantuan Bapak/Ibu, kami mengucapkan banyak terima kasih.

Hormat Kami,
Sekretaris Program Sarjana Magister

Reni Sonjaya, ST., MMSI

Jl. Salemba Raya No. 53 Jakarta Pusat 10440 Telp. (021) 3906518, 31902322, Jl. Kenari 13 Jakarta Pusat 10430 Telp (021) 31930220, 31930226, Jl. Margonda Raya No. 100 Pondok Cina, Depok 16424 Telp. (021) 78881112, 7863788 Ext. 403, Jl. Akses Kelapa Dua, Cimanggis Telp. (021) 8710561, 8727541 Ext. 106, Jl. KH. Noer Ali, Kalimalang Bekasi Telp. (021) 88860117, Jl. Stasiun Cakung Sentra Timur, Cakung Jakarta Timur Telp. 087788119797, Jl. Letjen TB Simatupang Kav. 28 Jakarta Selatan Telp. (021) 7801923, 7828418, Mutiara Taman Palem Blok C 7 No. 20, Jl. Raya Kamal Outer Ring Road, Cengkareng Telp. 087788119800, Jl. Akses Islamic Village, Kelapa Dua, Karawaci Telp. 087788119999, Jl. Cemara Raya No. 59 Jaka Sampurna Bekasi Barat, http://www.gunadarma.ac.id, e-mail:sekbor@gunadarma.ac.id

Figure A.2: Letter of data and information usage permission



MEKAR

PT Sampoerna Wirausaha

Sampoerna Strategic Square
North Tower Lt. 27
Jl. Jend. Sudirman Kav. 45-46
Jakarta Selatan 12930
Indonesia

P. 021 300 22735

SURAT KETERANGAN
No. 004/PTSW-INT.HR/I/2018

Berdasarkan surat **No. 006/Sek/Sarmag/UG/I/2018**, tentang permohonan penggunaan Data & Sistem Informasi, maka Pihak PT Sampoerna Wirausaha memberikan ijin kepada:

Nama	: **Alvian Dwi Kurnianto**
NPM	: **92216007**
Jurusan	: **Sistem Informasi Bisnis (Program SarMag)**

Untuk menggunakan Data & Sistem Informasi di PT Sampoerna Wirausaha untuk keperluan penyusunan Tesis.

Demikian surat keterangan ini kami buat dengan sebenarnya, dan untuk dipergunakan sebagaimana mestinya.

24 Januari 2018,
PT Sampoerna Wirausaha,

PT SAMPOERNA WIRAUSAHA

Dandi Rusli
CTO

www.mekar.id

CURRICULUM VITAE

PERSONAL INFORMATION

Name : Alvian Dwi Kurnianto

Birth Place and Date : Pekalongan, 4 September 1994

Sex : Male

NIM : 92216007

Address : Depok, West Java

Phone Number : 085840002746

Email : alviandk@gmail.com

FORMAL EDUCATION

2012 - 2016 : Information System Bachelor Degree at Gunadarma University

WORK EXPERIENCE

2016 - now : Mekar, Sampoerna Wirausaha - Remote Django Developer

2015 - 2017 : Coassets - Remote Django Developer

2014 - 2015 : Macroad - Django Developer Internship

2014 - 2015 : Coding Indonesia - Programming Tutor

SCIENCE PUBLICATION (2016)

2016 : Undergraduate Thesis titled “Animetacchi Website Development with Django and Ajax”